

Отчёта по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.

Барето Вилиан Мануел

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Реализация циклов в NASM	6
3.2	Обработка аргументов командной строки.	9
3.3	Задание для самостоятельной работы	12
4	Выводы	15

Список иллюстраций

3.1	Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code>	6
3.2	Заполняем файл	7
3.3	Запускаем файл и проверяем его работу	7
3.4	Изменяем файл	8
3.5	Запускаем файл и смотрим на его работу	8
3.6	Редактируем файл	9
3.7	Проверяем, сошелся ли наш вывод с данным в условии выводом .	9
3.8	Создаем файл командой <code>touch</code>	10
3.9	Заполняем файл	10
3.10	Смотрим на работу программ	10
3.11	Создаем файл командой <code>touch</code>	11
3.12	Заполняем файл	11
3.13	Смотрим на работу программы	11
3.14	Изменяем файл	12
3.15	Проверяем работу файла(работает правильно)	12
3.16	Создаем файл командой <code>touch</code>	13
3.17	Пишем программу	13
3.18	Смотрим на работу программы при $x_1=1$ $x_2=2$ $x_1=3$ (всё верно) . .	13
3.19	Смотрим на работу программы при $x_1=1$ $x_2=3$ $x_1=7$ (всё верно) . .	14

1 Цель работы

Изучить работу циклов и обработкой аргументов командной строки.

2 Задание

Написать программы с использованием циклов и обработкой аргументов командной строки.

3 Выполнение лабораторной работы

3.1 Реализация циклов в NASM

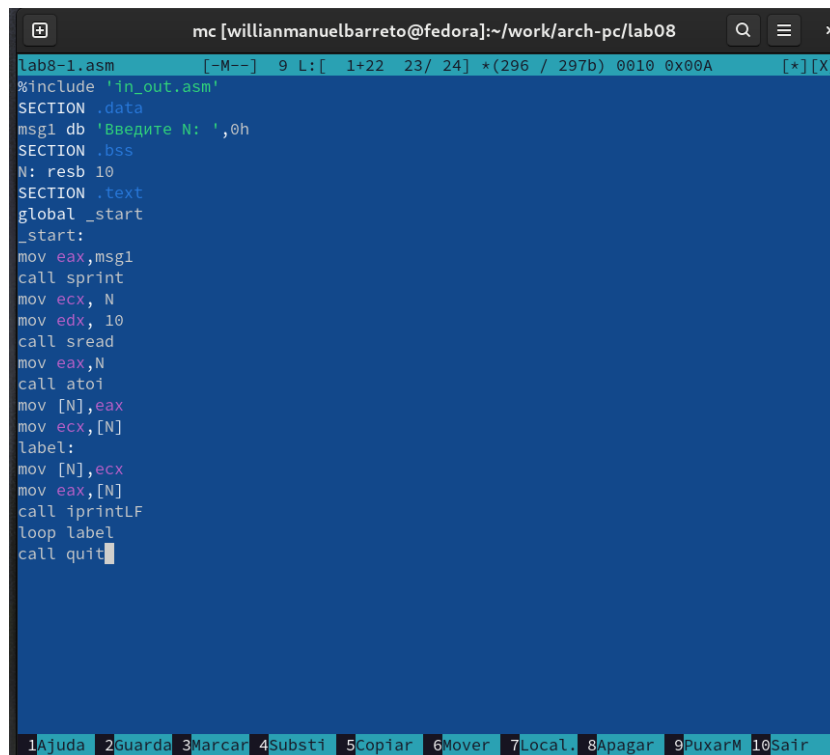
Создаем каталог для программ ЛБ8, и в нем создаем файл (рис. fig. 3.1).

A terminal window with a dark background. The title bar shows the user 'willianmanuelbarreto' and the current directory '~/work/arch-pc/lab08'. The terminal contains four lines of text: a prompt followed by 'mkdir ~/work/arch-pc/lab08', a prompt followed by 'cd ~/work/arch-pc/lab08', a prompt followed by 'touch lab8-1.asm', and a final prompt with a cursor. The text is in a light green color.

```
willianmanuelbarreto@fedora:~$ mkdir ~/work/arch-pc/lab08
willianmanuelbarreto@fedora:~$ cd ~/work/arch-pc/lab08
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ touch lab8-1.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

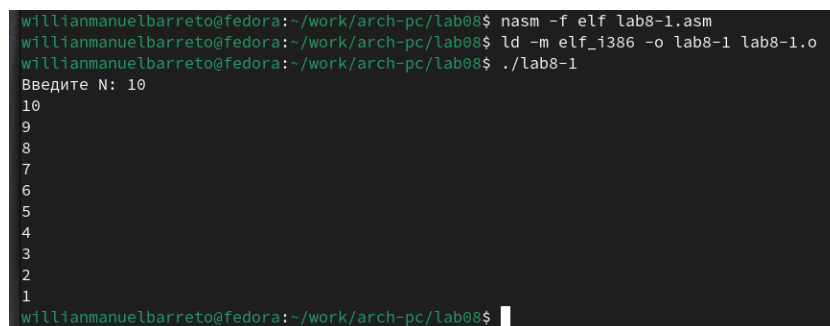
Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.1 (рис. fig. 3.2).



```
lab8-1.asm [-M--] 9 L: [ 1+22 23/ 24] *(296 / 297b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
mov eax,msg1
call sprint
mov ecx, N
mov edx, 10
call sread
mov eax,N
call atoi
mov [N],eax
mov ecx,[N]
label:
mov [N],ecx
mov eax,[N]
call iprintLF
loop label
call quit
```

Рис. 3.2: Заполняем файл

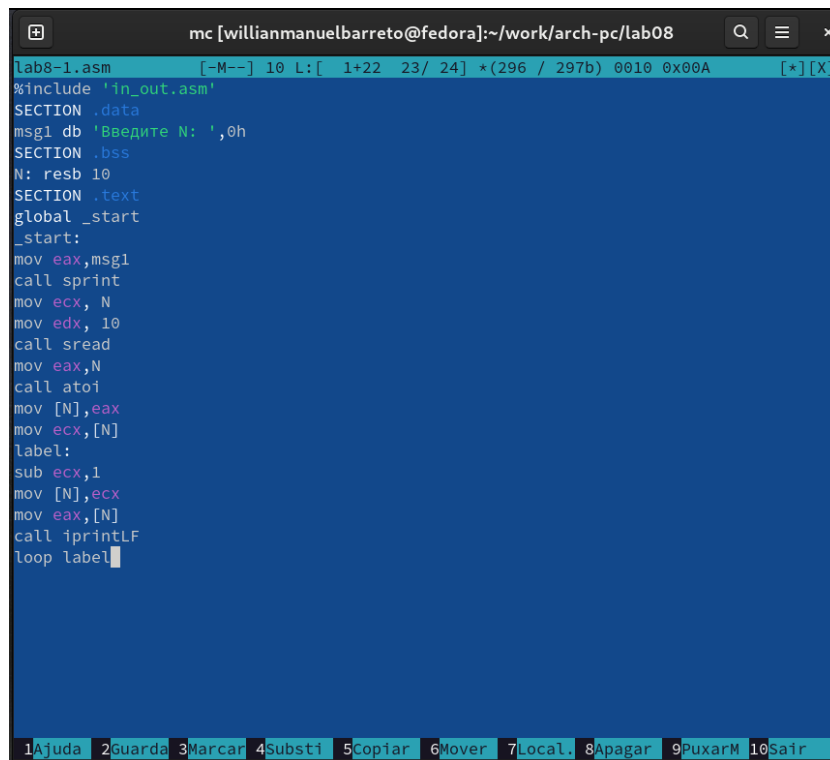
Создаем исполняемый файл и запускаем его (рис. fig. 3.3).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.3: Запускаем файл и проверяем его работу

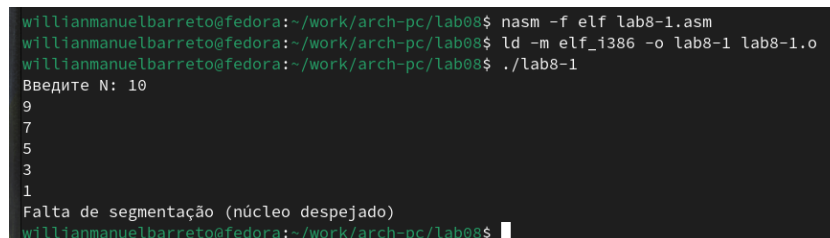
Снова открываем файл для редактирования и изменяем его, добавив изменение значения регистра в цикле (рис. fig. 3.4).



```
lab8-1.asm [-M--] 10 L: [ 1+22 23/ 24] *(296 / 297b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
mov eax,msg1
call sprint
mov ecx, N
mov edx, 10
call sread
mov eax,N
call atoi
mov [N],eax
mov ecx,[N]
label:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
loop label
```

Рис. 3.4: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. fig. 3.5).



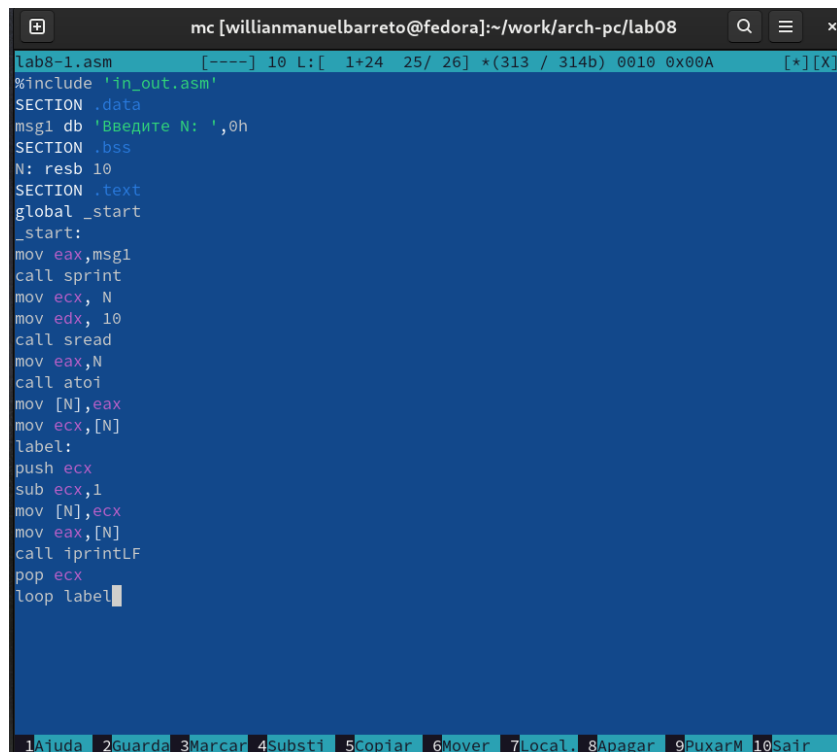
```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1
Falta de segmentação (núcleo despejado)
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.5: Запускаем файл и смотрим на его работу

Регистр ecx принимает значения 9,7,5,3,1(на вход подается число 10, в цикле label данный регистр уменьшается на 2 командой sub и loop).

Число проходов цикла не соответствует числу N, так как уменьшается на 2.

Снова открываем файл для редактирования и изменяем его, чтобы все корректно работало (рис. fig. 3.6).



```
lab8-1.asm [----] 10 L: [ 1+24 25/ 26] *(313 / 314b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
mov eax,msg1
call sprint
mov ecx, N
mov edx, 10
call sread
mov eax,N
call atoi
mov [N],eax
mov ecx,[N]
label:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx
loop label
```

Рис. 3.6: Редактируем файл

Создаем исполняемый файл и запускаем его (рис. fig. 3.7).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
Falta de segmentação (núcleo despejado)
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.7: Проверяем, сошелся ли наш вывод с данным в условии выводом

В данном случае число проходов цикла равна числу N.

3.2 Обработка аргументов командной строки.

Создаем новый файл (рис. fig. 3.8).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ touch lab8-2.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.8: Создаем файл командой touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.2 (рис. fig. 3.9).

```
lab8-2.asm [----] 9 L: [ 1+14 15/ 16] *(156 / 157b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
next:
cmp ecx, 0
jz _end
pop eax
call sprintf
loop next
_end:
call quit
```

Рис. 3.9: Заполняем файл

Создаем исполняемый файл и проверяем его работу, указав аргументы (рис. fig. 3.10).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-2.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ./lab8-2 1 2 '3'
1
2
3
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.10: Смотрим на работу программ

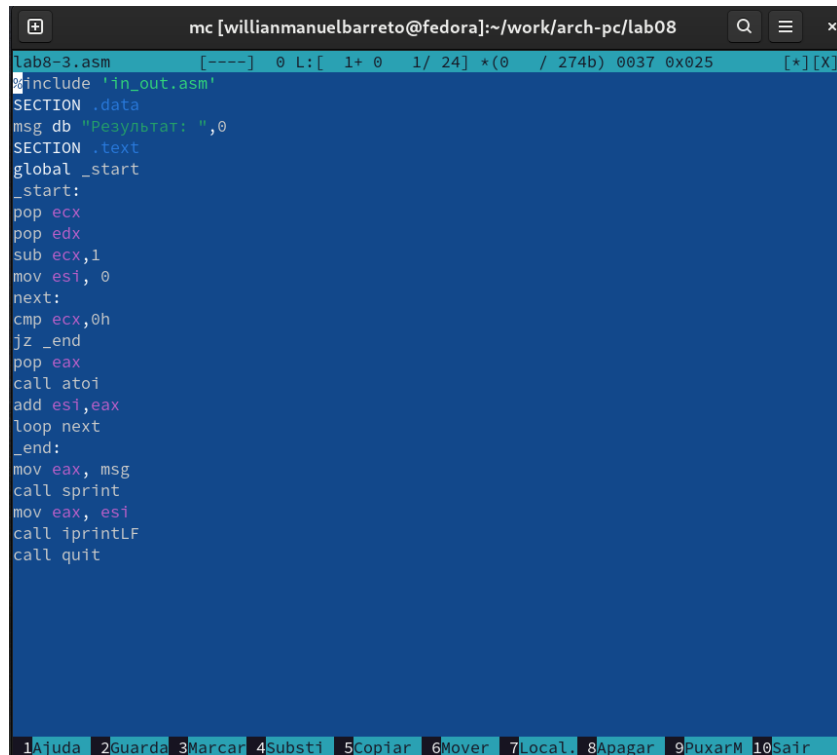
Программой было обработано 3 аргумента.

Создаем новый файл lab8-3.asm (рис. fig. 3.11).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ touch lab8-3.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.11: Создаем файл командой touch

Открываем файл и заполняем его в соответствии с листингом 8.3 (рис. fig. 3.12).



```
lab8-3.asm [----] 0 L: [ 1+ 0 1/ 24] *(0 / 274b) 0037 0x025 [*][X]
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi, 0
next:
cmp ecx,0h
jz _end
pop eax
call atoi
add esi,eax
loop next
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

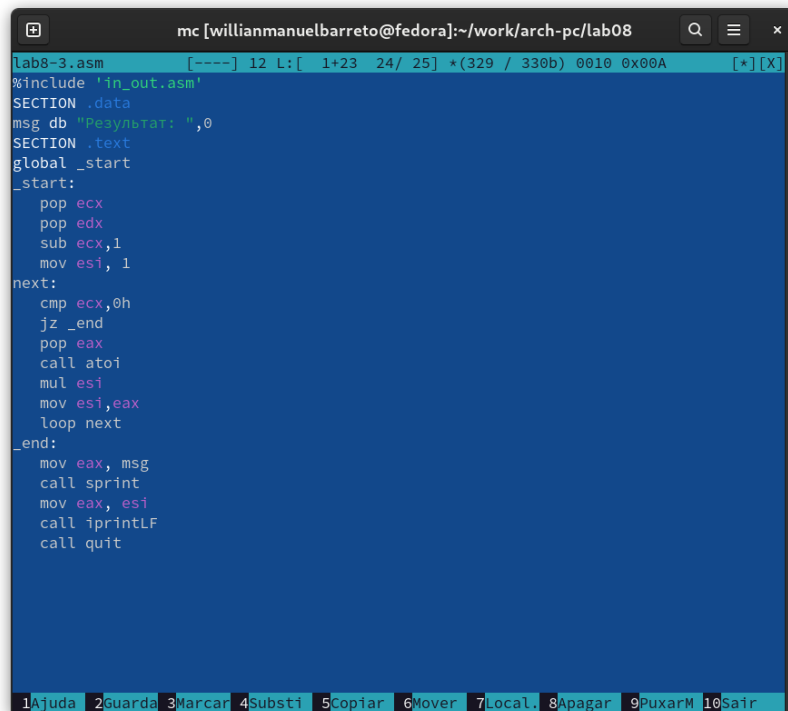
Рис. 3.12: Заполняем файл

Создаём исполняемый файл и запускаем его, указав аргументы (рис. fig. 3.13).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.13: Смотрим на работу программы

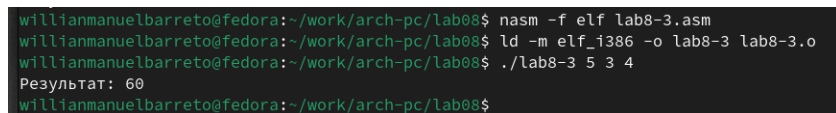
Снова открываем файл для редактирования и изменяем его, чтобы вычислялось произведение вводимых значений (рис. fig. 3.14).



```
lab8-3.asm [----] 12 L: [ 1+23 24/ 25] *(329 / 330b) 0010 0x00A [*][X]
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx
    pop edx
    sub ecx,1
    mov esi, 1
next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    mul esi
    mov esi,eax
    loop next
_end:
    mov eax, msg
    call sprint
    mov eax, esi
    call iprintLF
    call quit
```

Рис. 3.14: Изменяем файл

Создаём исполняемый файл и запускаем его, указав аргументы (рис. fig. 3.15).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$ ./lab8-3 5 3 4
Результат: 60
willianmanuelbarreto@fedora:~/work/arch-pc/lab08$
```

Рис. 3.15: Проверяем работу файла(работает правильно)

3.3 Задание для самостоятельной работы

ВАРИАНТ-20

1. Напишите программу, которая находит сумму значений функции $f(x)$ для $x = x_1, x_2, \dots, x_n$, т.е. программа должна выводить значение $f(x_1) + f(x_2) + \dots + f(x_n)$. Значения x_i передаются как аргументы. Вид функции $f(x)$ выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным

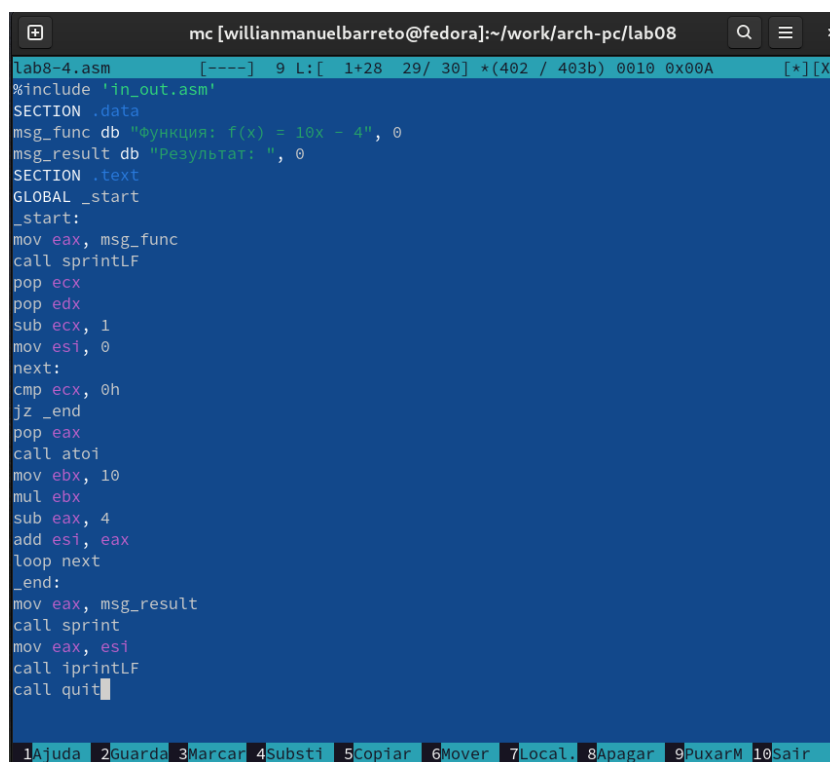
при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах $x = x_1, x_2, \dots, x_n$.

Создаем новый файл (рис. fig. 3.16).

```
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$ touch lab8-4.asm
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$
```

Рис. 3.16: Создаем файл командой touch

Открываем его и пишем программу, которая выведет сумму значений, полученных после решения выражения $10x-4$ (рис. fig. 3.17).



```
mc [willianmanuelbarreto@fedora]:~/work/arch-pc/lab08
lab8-4.asm [----] 9 L: [ 1+28 29/ 30] *(402 / 403b) 0010 0x00A [*] [X]
%include 'in_out.asm'
SECTION .data
msg_func db "Функция: f(x) = 10x - 4", 0
msg_result db "Результат: ", 0
SECTION .text
GLOBAL _start
_start:
mov eax, msg_func
call sprintf
pop ecx
pop edx
sub ecx, 1
mov esi, 0
next:
cmp ecx, 0h
jz _end
pop eax
call atoi
mov ebx, 10
mul ebx
sub eax, 4
add esi, eax
loop next
_end:
mov eax, msg_result
call sprintf
mov eax, esi
call iprintf
call quit
```

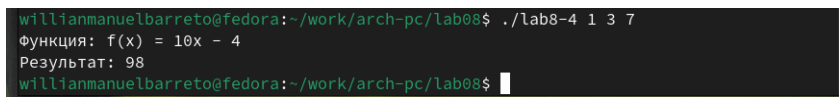
Рис. 3.17: Пишем программу

Транслируем файл и смотрим на работу программы (рис. fig. 3.18).

```
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$ ./lab8-4 1 2 3
Функция: f(x) = 10x - 4
Результат: 48
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$
```

Рис. 3.18: Смотрим на работу программы при $x_1=1$ $x_2=2$ $x_3=3$ (всё верно)

Транслируем файл и смотрим на работу программы (рис. fig. 3.19).

A terminal window with a dark background and green text. The prompt is 'willianmanuelbarreto@fedora: ~/work/arch-pc/lab08\$'. The command './lab8-4 1 3 7' has been entered. The output shows 'Функция: f(x) = 10x - 4' and 'Результат: 98'. The prompt is now 'willianmanuelbarreto@fedora: ~/work/arch-pc/lab08\$' with a cursor.

```
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$ ./lab8-4 1 3 7
Функция: f(x) = 10x - 4
Результат: 98
willianmanuelbarreto@fedora: ~/work/arch-pc/lab08$
```

Рис. 3.19: Смотрим на работу программы при $x_1=1$ $x_2=3$ $x_3=7$ (всё верно)

4 Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки.